DOCUMENT RESUME

ED 216 684 IR 010 162

AUTHOR Schulz, Russel E.; And Others
TITLE The Development of Programming Design Guides. Final Report.
INSTITUTION Human Resources Research Organization, Alexandria, Va.
SPONS AGENCY Army Research Inst. for the Behavioral and Social Sciences, Alexandria, Va.
REPORT NO ARI-RP-80-24a
PUB DATE Dec 79
CONTRACT DAHC19-78-C-0010
NOTE 28p.; For related documents, see IR 010 163-164, ED 203 864-868, and ED 205 202.

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS *Computer Oriented Programs; Flow Charts; Formative Evaluation; *Guides; *Instructional Development; Military Training; *Models; Programing Languages; *Systems Approach
IDENTIFIERS Authoring Aids; *Instructional Systems Development Model

ABSTRACT

The purpose of this research effort was to develop and evaluate a prototype system-independent Programming Design Guide (PDG) for one of the 13 offline job aids previously developed for use with the Instructional Systems Development (ISD) model. This PDG is intended to provide all of the guidance necessary for a user to implement the job aid on any of a large number of computer systems. The research effort was divided into three major tasks: (1) the establishment of the content and format for Programming Design Guides; (2) the development of a guide; and (3) the evaluation of the guide. Advice is given for developing Programming Design Guides for other blocks of the ISD model. Ten references are listed. (LLS)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED216684

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
This document has been reproduced as received from the person or organization originating it.
Minor changes have been made to improve reproduction quality.
Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

IR

RESEARCH PRODUCT 80-24a

THE DEVELOPMENT OF PROGRAMMING DESIGN GUIDES

BASIC SKILLS INSTRUCTIONAL SYSTEMS TECHNICAL AREA

DECEMBER 1979

IR010162

U.S. ARMY RESEARCH INSTITUTE for the BEHAVIORAL and SOCIAL SCIENCES

Approved for public release; distribution unlimited

U. S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency under the Jurisdiction of the Deputy Chief of Staff for Personnel

JOSEPH ZEIDNER
Technical Director

FRANKLIN A. HART
Colonel, US Army
Commander

NOTICES

FINAL DISPOSITION: This Research Product may be destroyed when it is no longer needed. Please do not return it to the U.S. Army Research Institute for the Behavioral and Social Sciences.

NOTE: This Research Product is not to be construed as an official Department of the Army document in its present form.

FOREWORD

The Computer-Based Instructional Systems Team of the US Army Research Institute for the Behavioral and Social Sciences (ARI) performs research and development in the area of educational technology that applies to military training. Of interest are methods for training individuals to develop and utilize instructional courseware in reasonable time, at acceptable cost.

This Research Product is one of a series which have been designed to support the implementation of the Instructional Systems Development Model (ISD, TRADOC Pamphlet 350-30). The ISD Model is a step-by-step procedure for the analysis, design, development, implementation, and control of military course materials. A previous effort produced manual Job Aids which are paper and pencil documents designed to provide "how to do it" guidance for the ISD Model. This document is part of a series of three developed to support the delivery of the manual Job Aids by computer. To accomplish this research, ARI's resources were augmented by contract DAHC19-78-C-0010 with the Human Resources Research Organization (HumRRO).

The contributions of personnel from ARI's Manpower and Educational Systems Technical Area as well as those of Mr. Charles F. Marshall and Mr. Joseph P. Severo, Research Facilities Support Group are acknowledged. Mr. Antonio J. Alameda, HumRRO also contributed to this research effort.

The entire research project is responsive to the requirements of Army Project 2Q263743A794, FY80 Work Program.

JOSEPH ZEIDNER
Technical Director

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Research Product 80-24a	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) THE DEVELOPMENT OF PROGRAMMING DESIGN GUIDES		5. TYPE OF REPORT & PERIOD COVERED Final Report January-October 1979
		6. PERFORMING ORG. REPORT NUMBER HumRRO RP-ED-79-9
7. AUTHOR(s) Russel E. Schulz, William G. Underhill, Carol S. Hargan, and Harold Wagner		8. CONTRACT OR GRANT NUMBER(s) DAHC19-78-C-0010
9. PERFORMING ORGANIZATION NAME AND ADDRESS Human Resources Research Organization 300 North Washington Street Alexandria, Virginia 22333		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 2Q263743A794
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Research Institute for the Behavioral and Social Sciences 5001 Eisenhower Avenue, Alexandria, Virginia 22333		12. REPORT DATE December 1979 13. NUMBER OF PAGES 19
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

This project was monitored technically by Dr. Harold F. O'Neil, Jr., Dr. Melissa Berkowitz, and Dr. Bruce W. Knerr.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

ISD
Instructional Systems Development
Author Aids
Job Aids
Programming Design Guide
Programming Design Language

20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The purpose of the research was to develop and evaluate a guide for the computer implementation of a manual Job Aid previously developed (ARI Research Products 80-13 and 80-14). The manual Job Aid of the prior research provided "how to do it" guidance for selected activities identified in the Instructional Systems Development Model (ISD, TRADOC Pamphlet 350-30). The present document describes the conduct of the research activities involved in creating a Programming Design Guide. Companion documents (ARI Research Products 80-24b and 80-24c) provide the Programming Design Guide and a supplemental handbook for users of the

012162

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

20. computer-based Job Aid.

THE DEVELOPMENT OF PROGRAMMING DESIGN GUIDES

BRIEF

Requirement:

The purpose was to develop a language to translate an existing paper and pencil Job Aid onto a computerized delivery system. The Job Aid is one of a series developed previously to support users of the Instructional Systems Development Model (ISD).

Procedure:

A Programming Design Language (PDL) was created to describe the computer functions (e.g., computer/user interactions, storage/retrieval of data, program branching, program management, and calculations) required by the Job Aids (ARI Research Products 80-13 through 80-18). The PDL was designed to communicate to the computer programmer in a language independent fashion so that the on-line or computer version of the Job Aid can be delivered by any computer.

Utilization:

The Programming Design Guide may be used by computer programmers who are tasked with programming the manual Job Aids.

TABLE OF CONTENTS

	Page
Background	1
Purpose	3
Research Activities	4
Task 1: Establish Content and Format for Programming Design Guides	4
Task 2: Develop the Programming Design Guide	7
Task 3: Evaluation of the Programming Design Guide	14
Conclusions and Recommendations	16
Guidance for Programming Design Guide Development	17
References	19

Figures

1	Facing Pages of Section VI of the Programming Design Guide	6
2	Page From Section II of the Programming Design Guide Describing the "SHOW" Command	9
3	Page From Section II of the Programming Design Guide Describing the "ITERATE" Command	10
4	Sample Section of Flowchart Contained in Section III of Programming Design Guide	11
5	Example of Variables Used in the Programming Design Guide	12
6	Example of "Setup Material" Contained in Section V of the Programming Design Guide	13

BACKGROUND

The systems approach to training defines a process which *focuses on the job* that is ultimately to be performed and *upon the individual* who is to learn to perform that job. The systems approach is just what the name implies: a systematic process for specifying the desired products of training and selecting what will be taught, how it will be taught, what the presentation mechanism will be, and evaluating the effects of each phase of the process. It focuses on student performance as a determinant of content. Its proper application can hardly fail to improve instruction where only incidental attention has been given to these functions. However, after more than two decades of research and development in this area, the full benefits of applying the systems approach to instruction have yet to be realized.

One attempt to standardize a definitive technology in this area is the Instructional Systems Development Model (ISD 1). The ISD contains standardized rationale, terminology, and basic concepts of instructional systems.

One of the primary difficulties of the ISD model and its manuals is that it emphasizes "what to do," not "how to do it." ISD manuals are intended to have general applicability. However, it has been clearly recognized by many experts that the same methodologies cannot be applied to the universe of training problems. "How to do it" may vary widely even though the specific processes or "what to do" may remain fairly constant across training problems. Thus, much specific guidance, tools, and procedures are needed by training designers, developers, and managers, for ISD to be appropriately implemented within the military training establishment.

In this report, "how to do it" guidance/tools/procedures will be referred to as author aids or job aids. An author is considered to be any member of an instructional staff charged with meeting the objectives of one or more components of the instructional development process.

A wide variety of aids exist, but the majority are more of the "what to do" than the "how to do it" variety. Rather than providing actual help in performing the authoring work, or even detailed how-to-do-it guidance, most of the existing author aids may serve simply to reinforce or broaden the guidance provided in ISD manuals. Care should be exercised in the selection of aids to be integrated with ISD so that authors are not confronted with a confusion of different yet similar models, sets of jargon, procedures or forms. Existing general manuals differ from one another in that they may:

- Include different steps or different names for steps.
- Include different methods of accomplishing each step.
- Provide different levels of specificity in the details included under each step.
- Provide different formats for reporting the work accomplished under each step.

Thus, to provide the most efficient aids to authors, the guidance found in some of these manuals and guidebooks needs to be translated and integrated into the ISD framework, rather than referred to in its source form.

Two author aids (2) developed in a previous project demonstrated the utility of our approach toward assisting authors implement the ISD process. The major characteristics of these aids which account for their usefulness are the following:

- The aids integrated existing ISD guidance.
- The aids were query-based and interactive.
- They were based upon a realistic authoring needs assessment at an operational site.
- Aids were developed and evaluated according to ISD guidance and recommended procedures.
- The aids concentrated on assisting authors dealing with structural and administrative details permitting them to concentrate their efforts on the subject matter content of the materials.

Current aids (3 through 8) cover the activities required for the first three phases of the ISD process. The 13 aids cover each block within these phases, excluding ISD Block I.1, Analyze Job. Each job aid is composed of two documents—a descriptive authoring flowchart and a job aid manual.

Descriptive Authoring Flowchart. The descriptive authoring flowchart is the primary document used in each job aid. It directs the user to specific guidance, examples and references provided in the job aid manual.

Job Aid Manual. The Job Aid Manual provides the specific guidance, examples and references necessary to produce the product for the instructional systems development activity it covers. In addition, each Job Aid Manual contains one or more worksheets to use in the development of the product.

At this stage, it is important to determine if and with what effort these off-line job aids can be converted into analogous on-line versions, regardless of the CBI system to be used. What guidance is needed by computer programmers within the Army to prepare programs that can implement these aids on their systems? These questions form the basis of the research effort described in this report.

PURPOSE

An earlier project (2) demonstrated the utility of on-line query-based author aids in the implementation of ISD Block II.2 (Develop Tests) and Block III.4 (Develop Instruction). The results of this study indicated the technical and practical merit of furthering the on-line, interactive aid concept to all applicable ISD Blocks and of providing these aids in a relatively system independent form. This has led to the research effort described in this report, the purpose of which is:

> to develop and evaluate a prototype system-independent Programming Design Guide (PDG) for one of the off-line author aids previously developed. This Programming Design Guide is intended to provide all of the guidance necessary for a user to implement the author aid on any of a large number of computer systems.

RESEARCH ACTIVITIES

The research effort was divided into three major tasks:

TASK 1: Establish content and format for Programming Design Guides

TASK 2: Develop a Programming Design Guide

TASK 3: Evaluation of a Programming Design Guide

Task 1: Establish Content and Format for Programming Design Guides

Task 1 activities were conducted in six steps which are described below.

Step 1. Study Existing Programming Design Languages

Programming Design Guides (PDG) are written in a Programming Design Language (PDL). Documents relevant to the development of programming design languages were examined to determine their usefulness for the PDL to be used in the research effort. None of these documents provided the needed guidance and so a suitable PDL had to be developed.

Step 2. Analyze Job Aid Flowcharts for Interactive Processes

In Step 2 the Descriptive Authoring Flowchart documents for the 13 author aids were carefully examined to identify all interactions between the user and the aids. The purpose of this examination was to provide greater assurance that the Programming Design Language would encompass the interactions required by any individual aid.

Step 3. Analyze Narrative Description of Job Aids for Additional Programming Design Language Requirement

The narrative description provided in the job aid manuals for each of the 13 author aids was examined to identify any additional programming design language requirements not obvious in the flowcharts. No additional programming design language requirements were identified as a result of this examination.

Step 4. Define/Describe Computer Functions Required for Job Aids

Computer functions such as computer/user interactions, storage/retrieval of data, program branching, program management, and calculations required for the aids were defined and described to the developers of the Programming Design Language. The identification in Steps 2 and 3 of the interactive processes within aids provided input to the definition of the computer functions.

Step 5. Develop Programming Design Language to Satisfy Functional Requirements

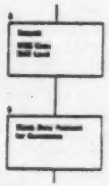
The basic requirement of the Programming Design Language is that it be similar to natural language so as to facilitate communication between designers and the computer programmers of the author aids independent of the computer system and programming language used. The Programming Design Language is not a programming language. It is a pseudo-computer language which is used to describe the design specification for certain classes of interactive computer programs. It was assumed that programmers using the Programming Design Guides would be skilled in an appropriate programming language. After completion of programming the resulting on-line version of the job aid is to be used by military instructional development personnel to perform the ISD activities required in the aid. PDL is designed to communicate to the programmer in a language independent fashion.

Step 6. Determine Format for the Programming Design Guide

The activities in the previous steps of Task 1 enabled us to determine the optimal format for the Programming Design Guide. The format developed provides for display of specific block(s) of the flowchart, the labels, commands, tags, and comments necessary for the programming of the author aid. An example of the format that was developed and used is shown in Figure 1.

Figure 1

Facing Pages of Section VI of the Programming Design Guide



BLOCK	COMMAND	TAG	COMMENTS
8a	SHOW	What MOS are you working with?	
	$ACCEPT	$MOSCODE,10	
	SHOW	What is the skill level?	
	ACCEPT	SKILL,1,4	
	SHOW	You are working with MOS /$MOSCODE/ and the skill level is /SKILL/. Is this correct?	
	DECIDE	block9a, block8a, block6a	
9a	SHOW	(text9a)	
	WAIT		
9b	ITERATE	DP , 1 , NUMBER_OF_DUTY_POSITIONS	
	SHOW	Duty position /DP/ is /$DUTYCOD*(DP)/	***roll on all duty position designators onto terminal. The end product should be a list of duty position designators.
	NEXT	DP	
	SHOW	Are these duty positions correct?	
	DECIDE	block10a, block9c	
9c	SHOW	(text9c)	
	WAIT		

(text9a) In previous ISD Blocks it was established that there are /NUMBER_OF_DUTY_POSITIONS/ duty positions.

/PRESS NEXT/ to see a list of the duty positions that were recorded in earlier ISD Blocks.

(text9c) It is extremely important that before you make any additions, deletions, or changes in duty positions that you check with:

- Your supervisor
- The individual(s) who prepared the Critical Task List (ISD I.2)
- The individual(s) who performed task analysis (ISD I.3). If they agree with your suggested additions, deletions, o changes you will be allowed to enter them into the termin.

Task 2: Develop the Programming Design Guide

The activities in Task 2 were conducted in five steps as described below.

Step 1. Select Aid for Programming Design Guide Development

In this step a specific job aid was selected for which the Programming Design Guide was to be developed. The job aid for ISD Block I.5 Select Instructional Setting was selected because of it is representative of the other aids in terms of function requirements, complexity, and difficulty in Programming Design Guide development. Approval of this selection was then obtained from the Army Research Institute's contracting officer's representative.

Step 2. Translate Job Aid Into Programming Format

Step 2 involved the three activities described below.

- Revision of the Flowchart for the job aid for ISD Block I.5 Select Instructional Setting. The flowcharts as currently presented in the author aids are at a level of detail sufficient to communicate off-line with military instructional development personnel. They require the instructional developer to provide and record information that can automatically be made available in computer versions of the author aids. Therefore, one of the activities in Step 2 was to revise the flowchart of the ISD I.5 aid for its intended use as part of the Programming Design Guide. In the revision process, care was taken to insure that the basic intent of the original flowchart was not affected.

- Data Storage/Retrieval. In this activity we identified the type of data that would normally be available to individuals charged with the ISD activity of selecting the instructional setting for each critical task, the type of data that would be generated as a result of the process of selecting the instructional setting, and an identification of where this collected data would be required in conducting the ISD activities of later ISD Blocks. We also identified when the data should be made available to the user and the format for its presentation. For example, in selecting instructional settings for critical military tasks it is necessary to have a list of these critical tasks and the duty positions within the MOS in which they are performed. The critical task listing and the duty positions comprising the MOS are determined in earlier ISD blocks. If these earlier blocks had been incorporated into on-line versions that information would already be in the computer. The Programming Design Guide would then need only to provide guidance in methods of accessing this information. However, since the information was not already resident in the computer, the Programming Design Guide provides guidance for placing the necessary information from earlier ISD blocks into the computer. The Programming Design Guide also makes provision for permanent storage of data, such as the instructional setting selected for each task, which would be used in later ISD blocks.

- Identification and Development of Basic Displays. Basic displays include materials such as a discussion of the purpose of a particular activity, techniques for performing the activity, sources of information, and questions asked the user. The display format that is used depends upon the specific display to be presented and its purpose. In addition, a display may require an active response from the user (answering a question). Because of the display limitations of the typical computer display devices, a decision was made that no display identified in the Programming Design Guide would be greater than 10 lines with a maximum of 40 alphanumeric characters per line. In addition, it was decided that since

many computer display devices do not have graphic capabilities, that all tables in the printed (off-line) version of the author aid would be presented in narrative form in the Programming Design Guide displays. All basic displays were thus identified in this step, and the format for their presentation to the user of the on-line author aid was established.

Step 3. Create Programming Specifications in Program Design Language

The expanded flowchart, data storage/retrieval requirements, and basic displays identified or developed in Step 2 were translated into programming specifications in terms of the Programming Design Language. These were prepared and incorporated into the Programming Design Guide.

Step 4. Prepare Draft Programming Design Guide

The Programming Design Guide (9) is organized into six sections as described below.

SECTION I: Introduction

SECTION II: Programming Design Language. This section describes the commands used in the Guide and provides guidance and examples of how each is used. Figures 2 and 3 show two pages of Section II of the Programming Design Guide. These pages describe the "SHOW" and the "ITERATE" commands.

SECTION III: Programming Flowchart. The flowchart included in Section III is an overview of the programming requirements for the entire program. Figure 4 shows a sample section of the flowchart contained in Section II of the Programming Design Guide.

SECTION IV: Variables Used in the Programming Design Guide. This section provides an alphabetical listing of all of the variables used in the program for implementation of the job aid for Selecting Instructional Setting. Figure 5 shows the first page of Section IV.

SECTION V: Setup Material. In this section guidance is provided for setting some variables to predetermined values and establishing various strings and arrays. The first page of Section V of the Programming Design Guide is shown in Figure 6.

SECTION VI: Programming Specifications. Section VI is the heart of the Programming Design Guide. As previously stated, it contains specific block(s) of the flowchart, the labels, commands, tags, and comments necessary for the programming of the Select Instructional Setting job aid. Figure 1 shows two facing pages of Section VI.

Step 5. Prepare Supplemental Guide

In addition to the Programming Design Guide, a hard copy supplemental guide was prepared. This supplement (10) contains textual material from the job aid that permits users of the on-line aid to review previously seen material.

Figure 2

Page From Section II of the Programming Design Guide Describing the "SHOW" Command

Implementation of Specific Text

Since computer systems differ among installations, the PDL includes the ability to specify where a PDL phrase should be consistently replaced with a phrase appropriate for a specific computer system. The PDL phrase is enclosed in a pair of # signs.

Example

```
SHOW      #PRESS NEXT# TO CONTINUE

#PRESS NEXT#  means that the user signals readiness to proceed
              by pressing a function key or typing a command.
```

In one implementation, this message might be "PRESS CARRIAGE RETURN to continue."

For another system, the message might say: "HIT CARRIAGE RETURN to continue."

Figure 3

Page From Section II of the Programming Design Guide Describing the "ITERATE" Command

```
ITERATE   index, first value, last value [,increment]
 .
 .
 .
```

The ITERATE command specifies the beginning of an iterative loop structure. The variable "index" is first set to the value "first value." Subsequent commands are processed in the normal manner, until a "NEXT index" command is encountered. The loop is then restarted with the variable "index" first having the value of "increment" added to it. The increment value defaults to one if unspecified. The loop terminates when the value of "index" becomes greater than "last value." Control then transfers to the next command after "NEXT index."

Examples

```
ITERATE  I , 1 , 10
.
.
.
NEXT  I

ITERATE J , 1 , 9 , 2          ****J will be 1, 3, 5, 7 and 9
.                                  during five loop passes.
.
.
NEXT J
```

Figure 4

Sample Section of Flowchart Contained in Section III of Programming Design Guide

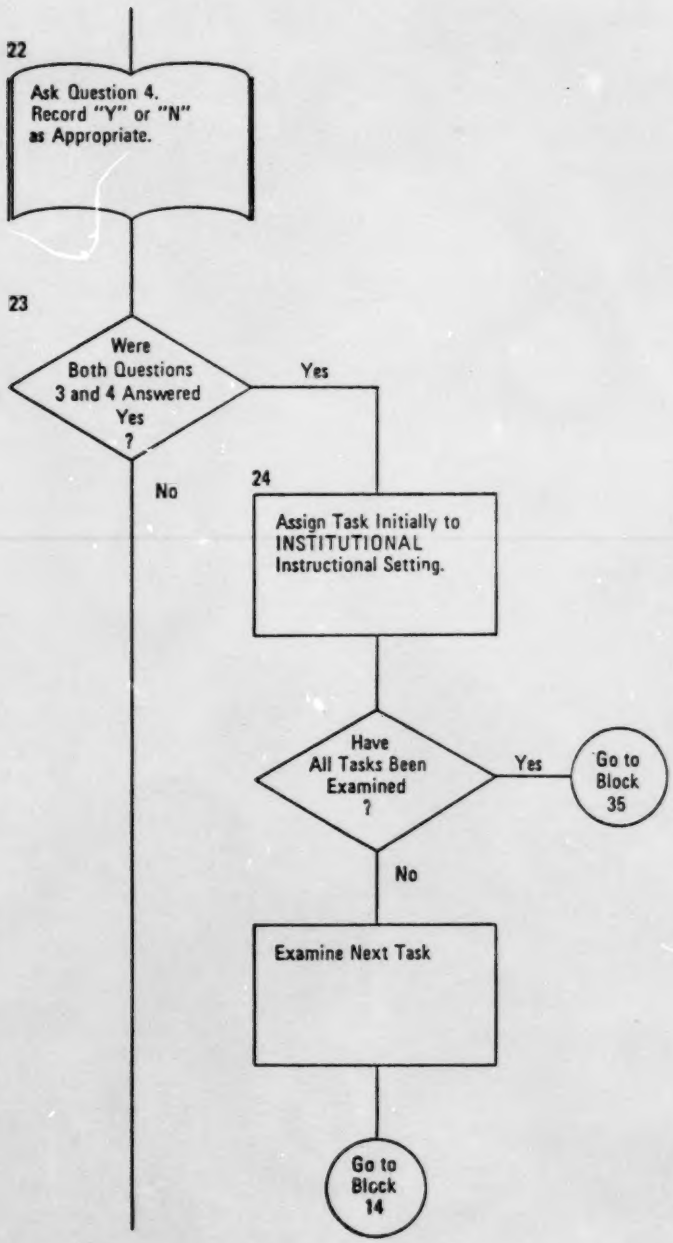


Figure 5

Example of Variables Used in the Programming Design Guide

Section IV

VARIABLES USED IN THE PROGRAMMING DESIGN GUIDE

This section of the Programming Design Guide provides an alphabetical listing of variables used in the program. Generally, the variable names are self-explanatory. However, where needed, an explanation of what the variable is used for is provided.

Keep in mind that these variable names are used only to communicate with programmers who are using the Programming Design Guide. Feel free to rename any variable.

DP	A numeric variable used to index the duty position array.
DPNEW	A numeric variable used to count the number of duty positions during the modification process.
$DUTYCODE	An array of alphanumeric variables used to define the duty positions.
$DUTYCODE2	A temporary array of alphanumeric variables used in the program when the user is modifying the duty position designations.
DUTYCODE_MAXIMUM	A numeric variable which defines the maximum number of alphanumeric characters that can make up a duty position title.
$ESTIMATE	Temporary variable that stores yes/no indicating whether percentage performing task is an estimate or not.
FIRST_TIME	Variable used to determine if the first unassigned task is being re-evaluated.
FSETTING	An array of numeric variables used to store the final designations of instructional settings. The numbers stored will be a 1, 2, or 3.
HP_CRITERION	Criterion value for percentage of soldiers who must perform a task before the task is identified as a "high performance task."
$INPUTLINE,100	An alphanumeric variable that is to store temporarily the user's input duty code designation. It is limited to 100 characters.

Figure 6

Example of "Setup Material" Contained in Section V of the Programming Design Guide

Section V

SETUP MATERIAL

To facilitate the programming of the computer version of the Job Aid for Selecting Instructional Settings, it is necessary that you first program (in your programming language) setup material. A guide for the necessary setup material is shown below. You must, of course, establish your own value for MAXIMUM TASKS, MAXIMUM_DUTY_POSITIONS, and NUMBER_OF_DUTY_POSITIONS. In addition, you will provide your own task ID numbers and task titles for $TASKCODE(1) thru $TASKCODE(n) -- limit 40 characters -- and your duty positions for $DUTYCODE(1) thru DUTYCODE(n) -- limit 30 characters.

```
SET       MAXIMUM_TASKS = 24
SET       MAXIMUM_DUTY_POSITIONS = 15
SET       TASKCODE_MAXIMUM = 40
SET       DUTYCODE_MAXIMUM = 30
$STRING   $TASKCODE(MAXIMUM_TASKS), TASKCODE_MAXIMUM
$STRING   $DUTYCODE(MAXIMUM_DUTY_POSITIONS), DUTYCODE_MAXIMUM
$STRING   $DUTYCODE2(MAXIMUM_DUTY_POSITIONS,DUTYCODE_MAXIMUM)
$STRING   $MOSCODE,10
ARRAY     ISR_DUTY(MAXIMUM_DUTY_POSITIONS,MAXIMUM_TASKS)
ARRAY     ISR%(MAXIMUM_TASKS)
ARRAY     ISR%_E(MAXIMUM_TASKS)
ARRAY     SETTING(MAXIMUM_TASKS)
ARRAY     FSETTINGS(MAXIMUM_TASKS)
ARRAY     ISR_QUESTION(14,MAXIMUM_TASKS)
$STRING   $INSTR_SETTING(3),20
$SET      $INSTR_SETTING(1) = "INSTITUTIONAL"
$SET      $INSTR_SETTING(2) = "S O J T"
$SET      $INSTR_SETTING(3) = "SELF STUDY"
SET       NUMBER_OF_TASKS = 24

$SET      $TASKCODE(1) = 574-2058    Operate Radio Test Set AN/VRM-1 to
                                     Test Modules in AN/VRC-12 Series
                                     Radio Sets

$SET      $TASKCODE(2) = 587-0025    Repair Radio Set, AN/PRC-25/77

$SET      $TASKCODE(3) = 587-0032    Systems Troubleshooting Radio Set,
               .                     AN/VRC-12 including C-2742/VRC to a
               .                     Defective Component, Cable or
               .                     Accessory
$SET      $TASKCODE(n) = 587-1027    Verify installation of Radio Set
                                     AN/VRC-46 in a Tracked Vehicle
```

Task 3: Evaluation of the Programming Design Guide

In Task 3 we formatively evaluated and revised the Programming Design Guide developed in Task 2. The purpose of this Task was to obtain whatever information we could to determine and improve the usefulness and generality of the Programming Design Guide.

The evaluation was conducted in three steps. The first two steps involved the installation of the Job Aid for Selecting Instructional Setting into two different Computer-Based Instruction (CBI) systems using the Programming Design Guide. In the third step the Programming Design Guide was indirectly evaluated by using the resulting on-line program to simulate selection of instructional settings for a sample of military tasks. A description of each evaluation step and its outcome follows.

Step 1. Formative Evaluation

- Procedure. As major portions of the programming specifications in the draft Programming Design Guide were completed, a project staff member used the Guide to program the material into a Digital Group Z-80-based micro-computer with video display. The programming design language in the Guide was translated into BASIC for computer implementation. The purpose of this evaluation was to identify major deficiencies in the Programming Design Guide. Because of limited memory capabilities of the micro-computer, narrative displays were not produced in their entirety.

- Results of the Evaluation. The programmer had no difficulty in using the Programming Design Guide. The programmer was thoroughly familiar with the author aid covered by the Programming Design Guide but had not been directly involved in the preparation of the program specifications included in the Guide. This evaluation resulted in the identification of minor "bugs" in the Programming Design Guide which were immediately corrected. It required approximately 100 hours to program the aid in BASIC using the Programming Design Guide.

Step 2. Formative Evaluation by U.S. Army Research Institute Systems Analyst

- Procedure. After each portion of the programming specifications (Section VI, Programming Design Guide) had been programmed into the micro-computer and revisions made, itwas sent to the U.S. Army Research Institute. A systems analyst was provided to install the aid using the Programming Design Guide.[1] This systems analyst was in no way associated with the development of either the Programming Design Language or the Programming Design Guide. The computer system used for installing the aid was a CDC 3300 computer and a CDC 210/11 CRT. The CRT has a 50 x 20 matrix screen which is limited to 1000 characters. It produces only upper case letters in white on a black screen. In this evaluation the Programming Design Language was translated into FORTRAN.

[1] The ARI systems analyst was Charles F. Marshall.

The specific procedure used in this step of the evaluation was as follows:

— The systems analyst programmed the job aid into the computer using only the Programming Design Guide (without further assistance).

— An anecdotal record was maintained.

— The time taken to program the aid was recorded.

— Telephonic communication was maintained between HumRRO and the systems analyst on almost a daily basis to discuss progress.

• Results of the Evaluation. No major problems arose in the installation of the job aid on the computer using the Programming Design Guide. The systems analyst did not always agree with the wording in some of the narrative displays shown in the Programming Design Guide. In a few cases he asked to change the wording. This was permitted unless it changed the intent of the narration. However, no programming problems occurred that required a major revision of the Programming Design Guide.

After each portion of the aid had been installed into the computer the systems analyst and the project director reviewed the resulting program on-line to identify any additional problems resulting from an incorrect interpretation of the Programming Design Guide. No major problems were discovered as a result of this review. Approximately 115 hours were required for the systems analyst to install the job aid on the CDC 3300 computer. This time involved the following activities:

— Translating Programming Design Language into FORTRAN	36 hours
— Keypunching and assembling decks	18 hours
— Debugging and revisions	61 hours
TOTAL	115 hours

Step 3. Formative Evaluation of On-line Aid

• Procedure. After the entire Job Aid for Selecting Instructional Setting had been installed into the CDC 3300 computer, and had been "debugged," a final evaluation of the on-line program was conducted. This evaluation was performed by the alternate ARI contracting officer's representative.[1] This individual used the on-line program in much the same way as would an intended military user, that is, a simulated task to select instructional settings was performed by a COR. The purpose of this part of the evaluation was to identify any additional "bugs" in either logic or narrative material.

• Results of Evaluation. Only minor changes in wording of narrative material were required as a result of this evaluation effort. The evaluation required approximately five hours to complete.

[1]The alternate ARI COR was Dr. Melissa Berkowitz.

CONCLUSIONS AND RECOMMENDATIONS

The following are conclusions and recommendations gleaned from the formative evaluation of the Programming Design Guide.

- The Programming Design Guide for the Select Instructional Setting job aid was an easily used and effective tool for installing this aid into two widely different computer systems. Neither the project programmer nor the systems analyst encountered any difficulty in translating the Programming Design Guide into a meaningful and effective on-line program. The time required for installing the job aid on the computer using the Programming Design Guide was 100 hours for the project programmer. It required 115 hours for the systems analyst to install the aid on the computer. In neither case is the time for installation of the aid on the computer excessive. They are probably representative of the time that would be required for installation on other computer systems.

- The logic contained in the on-line programming of the job aid is complex. A misinterpretation of the guidance contained in the Programming Design Guide can result in an error in the on-line program which might not be readily apparent to the programmer. For this reason it is strongly suggested that detailed guidance is needed for checking the accuracy of the programmed, on-line job aids. This guidance should be in the form of leading the "debugger" through each step of the program, stating exactly what to enter and the result of each entry. In our opinion, a separate "debugging" guide is required for each Programming Design Guide.

- We determined that the on-line version of the author aid produced by using the Programming Design Guide was an effective instructional systems development tool for selecting instructional settings for training military tasks. There is no apparent reason why Programming Design Guides developed for other ISD blocks would not be equally effective. Therefore, it is recommended that Programming Design Guides be developed for the remaining blocks of the ISD model.

- The guidance provided in the next section should be sufficient to develop Programming Design Guides for any specific ISD block. However, the effort required to develop any PDG is considerable. Certainly it would not be cost-effective for every instructional systems development installation to develop their own Programming Design Guides for the ISD blocks. A master set should be developed covering all of the ISD blocks and these should be distributed to the appropriate ISD installations for implementing author aids on their own computer systems.

GUIDANCE FOR PROGRAMMING DESIGN GUIDE DEVELOPMENT

As stated in the preceding section of this report, we feel that the present research activity indicates the desireability of developing Programming Design Guides for other blocks of the ISD model. Further development of Programming Design Guides will not require all of the steps performed in the initial development activity. In our opinion, only the following steps need to be performed when developing Programming Design Guides for the remaining blocks.

Step 1. Analyze Job Aid Flowcharts for Interactive Processes

Step 1 involves the careful study of the author aid to obtain an overview of the requirements of the Programming Design Guide to be developed.

Step 2. Translate Job Aid Into Programming Format

- Revision of Flowchart. The flowchart contained in the printed (off-line) version of the author aid should be studied and revised as necessary to provide sufficient assistance for input into a computer system.

- Data Storage/Retrieval. In this activity the data that should be available to the user of the resulting on-line program should be identified and provision made within the Programming Design Guide for making the data available. In addition, the type of data to be generated within the program should be identified and specific points within the program at which it should be made available should also be identified. Provisions for storage and retrieval of the data should be included in the Programming Design Guide.

- Identification and Development of Basic Displays. Basic displays of materials such as a discussion of the purpose of a particular activity, techniques for performing the activity, sources of information, questions asked the user, and other information specific to the author aid for which the Programming Design Guide is being developed should be identified for use in step 3.

Step 3. Create Programming Specifications in the Programming Design Language

The revised flowchart, data storage/retrieval requirements, and basic displays identified or developed in Step 2 are translated into programming specifications in the Programming Design Language for inclusion in the Programming Design Guide.

Step 4. Prepare Draft Programming Design Guide.

Using the information identified or developed in Steps 1 through 3 a draft version of the Programming Design Guide is prepared in Step 4. The Programming Design Guide should be organized into six sections similar to those in the Guide developed in this research effort. The sources of information for each section is as follows:

SECTION I: Introduction. Use the introduction in the original Guide.

SECTION II: Programming Design Language. All Programming Design Language commands included in the present Guide should be employed.

SECTION III: Programming Flowchart. Specific flowchart blocks would be derived from Step 2 above. However, the format should be the same as that in the present Guide.

SECTION IV: Variables Used in the Programming Design Guide. These variables would be developed in Step 3.

SECTION V: Setup Material. Essentially the same format as in the present Guide. Specific variables that must be set to predetermined values and the various strings and arrays that need to be established would be identified in Step 2.

SECTION VI: Programming Specifications. This is to be obtained from activities in Step 3.

Each of the resulting Programming Design Guides should be ac ompanied by a guide for "debugging" the on-line program. To provide a standardized format nomenclature, and methodology for implementing the various Programming Design Guides, it is strongly recommended that the complete set be prepared at a central facility.

REFERENCES

1. Interservice Procedures for Instructional Systems Development, TRADOC Pamphlet 350-30 (Phases I through V and Executive Summary), August, 1975.

2. Schulz, R.E., Hibbits, N., Wagner, H. and Seidel, R.J. *On-line Authoring Aids for Instructional Design*, HumRRO Final Report FR-ED-77-24, October, 1977.

3. Job Aids: Descriptive Authoring Flowcharts for Phase I - Analyze, of the Instructional Systems Development Model, Research Product, U.S. Army Research Institute, in press.

4. Job Aids: Descriptive Authoring Flowcharts for Phase II - Design, of the Instructional Systems Development Model, Research Product, U.S. Army Research Institute, in press.

5. Job Aids: Descriptive Authoring Flowcharts for Phase III - Develop, of the Instructional Systems Development Model, Research Product, U.S. Army Research Institute, in press.

6. Job Aid Manuals for Phase I - Analyze, of the Instructional Systems Development Model, Research Product, U.S. Army Research Institute, in press.

7. Job Aids Manuals for Phase II - Design, of the Instructional Systems Development Model, Research Product, U.S. Army Research Institute, in press.

8. Job Aids Manuals for Phase III - Develop, of the Instructional Systems Development Model, Research Product, U.S. Army Research Institute, in press.

9. Schulz, R.E., Underhill, W.G., and Hargan, C.S. *Programming Design Guide for Computer Implementation of Job Aid for Selecting Instruction Settings*, HumRRO Research Product 79-10, December 1979.

10. Schulz, R.E. *Supplemental Guide: Sources of Information for On-line Implementation of ISD I.5 Select Instructional Setting*, HumRRO Research Product 79-9, December, 1979.